```
? s pn=01068853
S2          1    S PN=01068853

? t o s2/full
>>>E:  Set o s2 does not exist

? s pn=04117548
S3          1    S PN=04117548

? t s2/full
```

2/19/1 **Links**

02771253 **Image available**

MEMORY CONTROL SYSTEM

Pub. No.: 01-068853 [JP 1068853 A]
Published: March 14, 1989 (19890314)
Inventor: KANEKO TADASHI
SUDO KIYOSHI
Applicant: FUJITSU LTD [000522] (A Japanese Company or Corporation), JP (Japan)
Application No.: 62-227020 [JP 87227020]
Filed: September 10, 1987 (19870910)
International Class: [4] G06F-012/06
JAPIO Class: 45.2 (INFORMATION PROCESSING -- Memory Units)
Journal: Section: P, Section No. 892, Vol. 13, No. 284, Pg. 19, June 29, 1989 (19890629)

ABSTRACT

PURPOSE: To shorten the queuing time of a CPU, by checking the start of write at every stage of interleave and starting the read data transfer report to the CPU and a read data holding means after confirming that the stage is idle.

CONSTITUTION: An interleave control part 11 is provided with a read data holding means 13 where read data from memory banks 1-4 is held and a control means 12, and this control means checks the start of write it every stage of interleave and starts the read data transfer report to CPUs 5-8 and a read data holding means 13 after confirming that the stage is idle. Thus, when overlapping between transfer of read data and that of write data is forecasted, read data is held in the read data holding means 13 and its transfer can be delayed with one stage as the unit, and the queuing time of CPUs 5-8 is shortened as much as possible.

⑲日本国特許庁（JP）　⑪特許出願公開

⑫公開特許公報（A）　昭64－68853

⑤Int.Cl.⁴	識別記号	庁内整理番号	㊸公開　昭和64年(1989)3月14日
G 06 F 12/06		Q－8841－5B	

審査請求　未請求　発明の数 1　（全7頁）

㊹発明の名称　メモリ制御方式

㉑特　願　昭62－227020

㉒出　願　昭62(1987)9月10日

㉜発明者　金古　正　神奈川県川崎市中原区上小田中1015番地　富士通株式会社内

㉜発明者　須藤　清　神奈川県川崎市中原区上小田中1015番地　富士通株式会社内

㉛出願人　富士通株式会社　神奈川県川崎市中原区上小田中1015番地

㉔代理人　弁理士　井桁　貞一

明　細　書

1．発明の名称

メモリ制御方式

2．特許請求の範囲

(1)記憶部の各メモリバンク（1～4）に対するプロセッサ（5～8）からの並列アクセスを処理するインタリーブ制御部（11）を備えたメモリ制御方式において、

メモリバンク（1～4）からのリードデータを保持するリードデータ保持手段（13）と、

インタリーブの各ステージ毎にライトの起動をチェックし、ステージが空いていることを確認したのちにプロセッサ（5～8）へのリードデータ転送通知及び前記リードデータ保持手段（13）の起動を行う制御手段（12）とを、

インタリーブ制御部（11）が備えることを特徴とするメモリ制御方式。

(2)各プロセッサ（5～8）の優先度をインタリーブ制御よりも優先させるプロセッサ優先度処理部（15）を備えたことを特徴とする特許請求の範囲第1項に記載のメモリ制御方式。

3．発明の詳細な説明

〔概　要〕

本発明は、インタリーブ制御を行うデータ処理装置において、データの書込みと読出しとを同一のデータバスで行うメモリ制御方式に関し、

プロセッサ（以下、ＣＰＵと呼称する）の待ち時間を可能な限り短縮し、高速かつ効率的な処理を実行することを目的とし、

メモリバンクからのリードデータを保持するリードデータ保持手段と、インタリーブの各ステージ毎にライトの起動をチェックし、ステージが空いていることを確認したのちにＣＰＵへのリードデータ転送通知及び前記リードデータ保持手段の起動を行う制御手段とを、インタリーブ制御部が備えるように構成する。

〔産業上の利用分野〕

本発明は、少なくとも1個以上のＣＰＵと複数

個のバンクに分割されたメモリで構成され、インタリーブ制御を行うデータ処理装置のメモリ制御方式に関し、特に、データの書込みと読出しとを同一のデータバスにより時分割で行うメモリ制御方式に関する。

データ処理装置の主記憶等で、アクセスを高速化するため、メモリを複数のバンクに区分して、それらのメモリバンクに並列アクセスを行う方法が行われている。メモリ内のアドレスが連続しているとアクセスはシリアルになるので、アドレスの下位ビットをふり分けてメモリを区分したものがメモリバンクで、それらの並列アクセスを管理するために、交互処理を行うインタリーブ制御部が介設されるのが普通である。

〔従来の技術〕

通常、メモリバンクに対するアクセスラインはＣＰＵが複数であっても、１本のアドレスバスと１本のデータバスとで形成されていて、その１本のアドレスバスをインタリーブ制御部が制御している。

第５図は、従来のメモリ制御方式の一例を示す構成図である。第５図において、記憶部は４個のメモリバンク１～４で構成され、これらを４個のＣＰＵ５～８が使用するようになっている。記憶部とＣＰＵ側との間は、１本のアドレスバス９と１本のデータバス１０で接続されていて、それらのバスはメモリバンク１～４及びＣＰＵ５～８に並列に分岐接続されている。インタリーブ制御部１１はアドレスバス９上に配設されていて、アドレス信号と制御信号とを管理する。

上記のようなインタリーブ制御方式では、本質的にデータバス１０が１本なので、メモリバンクからＣＰＵへのリードデータの転送とＣＰＵからメモリバンクへのライトデータの転送とが重なる場合、いずれかのアクセスの起動を延期させなければならない。メモリとしては、読出しはリードストローブを受信すると直ちにデータをバスに乗せることができるが、書込みはクロックと同期したサイクルで行わなければならないので、一般的

3　　4

なインタリーブ制御では、リードデータの転送とライトデータの転送が重なると予想される場合、リードデータを優先させるようになっている。

〔発明が解決しようとする問題点〕

しかし、上記従来の方式では、ライトデータの転送が大幅に待たされることがあり、処理全体も極めて緩慢になる可能性がある。

第６図は、従来のインタリーブ制御の一例を示すタイミングチャートである。１サイクルとは、連続するリード／ライト周期動作の開始点間の最少時間間隔で、第６図においては、これをメモリバンクの数に対応する４つのステージに区切って各動作を処理するようになっている。但し、インタリーブ制御部からメモリバンクにアクセスする際、書込みの場合は次のステージで直ちにライトストローブ信号を起動できるが、読出しの場合はアドレスの指定等で、少なくとも３ステージ目でなければリードストローブ信号を起動できない。その結果、第６図に示すように、第１サイクルの初頭にＣＰＵ１及びＣＰＵ２からリードが起動されると、それらは第１ステージ（Ｓ１）及び第２ステージ（Ｓ２）に受付処理され、３ステージ後の第４ステージ（Ｓ４）及び第２サイクルの第１ステージ（Ｓ１′）でメモリバンクからそれぞれのデータを読み出されるが、同じ第１サイクル内にＣＰＵ３及びＣＰＵ４からライトが起動されたとしても、データバスをリードに優先させる設定に従って第２サイクルの第１ステージ（Ｓ１′）以後になり、しかも書込みはライトのサイクルに従うので、インタリーブ制御部としては第２サイクルの第３ステージ（Ｓ３′）と第４ステージ（Ｓ４′）にライトストローブを起動し、それぞれ次のステージでデータをバスに乗せることになる。このように、従来の方式では、第１サイクルでＣＰＵ４の起動したライトが、何と第３サイクルに持越されるという事態も起り得るほどで、全体的な処理が極めて緩慢になる。

本発明は、このような問題点に鑑みて創案されたもので、ＣＰＵ側の待ち時間を可能な限り短縮

5　　6

し、高速かつ効率的な処理を実現するメモリ制御方式を提供することを目的としている。

〔問題点を解決するための手段〕

本発明において上記の問題点を解決するための手段は、ＣＰＵから記憶部の各メモリバンクに対する並列アクセスを処理するインタリーブ制御部を備えたメモリ制御方式において、メモリバンクからのリードデータを保持するリードデータ保持手段と、インタリーブの各ステージ毎にライトの起動をチェックし、ステージが空いていることを確認したのちにＣＰＵへのリードデータ転送通知及び前記リードデータ保持手段の起動を行う制御手段とを、インタリーブ制御部が備えたメモリ制御方式によるものとする。但し、所要によっては、各ＣＰＵの優先度をインタリーブ制御よりも優先させるＣＰＵ優先度処理部を備えるものとする。

〔作　用〕

本発明では、インタリーブ制御部内の制御手段

7

により、メモリバンク１～４からインタリーブされて出力されるリードデータを先着順に保持し、同じく制御手段１２からのドライブ信号により、保持しているリードデータを前記先着順にデータバス１０に出力する。

制御手段１２は、ＣＰＵ５～８にライン９ａ及びライン９ｂで接続され、メモリバンク１～４にライン９ｃで接続されていて、各ＣＰＵ５～８からライン９ａで送られて来る制御信号，アドレス信号及びライトストローブ信号を受取り、その制御信号及びアドレス信号をライン９ｃで各メモリバンク１～４へ伝達し、逆に各ＣＰＵ５～８へはライン９ｂでリードストローブ信号を通知する。ライトストローブ信号はＣＰＵからメモリバンクへライトデータが転送されることを通知する信号であり、リードストローブ信号はリードを起動したＣＰＵを識別する情報を含み、ＣＰＵへリードデータがデータバスに出力されていることを通知する信号である。また、制御手段１２は、リードサイクルが起動される毎に、必要なステージで、

9

が、各ステージ毎にライトの起動をチェックし、リードデータの転送とライトデータの転送が重なると予想される場合、読み出したリードデータをリードデータ保持手段で保持し、その転送を１ステージ単位で待たせることができるようにして、ＣＰＵの待ち時間を極力短縮するものである。

〔実施例〕

以下、図面を参照して、本発明の実施例を詳細に説明する。

第１図は、本発明を実施したメモリ制御方式の一例を示す構成図である。前記従来例と同様に、第１図においても、記憶部は４個のメモリバンク１～４で構成され、これらを４個のＣＰＵ５～８が使用するようになっていて、記憶部とＣＰＵ部との間にはインタリーブ制御部１１が介設されている。インタリーブ制御部１１は、制御手段１２とリードデータ保持手段１３とを内蔵している。

リードデータ保持手段１３は、データバス１０上に配設され、制御手段１２からのクロック信号

8

リードデータ保持手段１３へクロック信号を出力し、メモリバンクからのリードデータを保持させると共に、ＣＰＵからのライトストローブ信号を受取って、下記の判断と動作を行う。

第２図は、上記実施例のインタリーブ制御部の動作手順を示すフローチャートである。第２図において、制御手段１２は、まずフローの第１段で制御信号，アドレス信号及びライトストローブ信号を受取ると、現在のアクセスがリードであるかライトであるかチェックする。ライトであれば、次のステージでそのままメモリライトを開始し、ライトデータを受信して、２ステージ後にメモリ素子にデータを書込めばよい。

アクセスがリードの場合は、フローの第３段として、アドレス指定されたメモリのリードを次のステージで開始し、フローの第４段として、２ステージ後に、リードデータ保持部１３にクロック信号を出力してメモリからのリードデータを保持させる。

ここで、制御手段１２は１ステージ前にライト

10

ストローブを受信していないか起動をチェックしてフロー第５段の判断を行い、ＣＰＵからライトストローブがアサートされていなければ、リードデータの転送とライトデータの転送が同一ステージで重ならないものとして、フローは下方へ分岐し、３ステージ後、制御手段１２はリードデータ保持部１３にドライブ信号を出力し、かつＣＰＵへリードストローブ信号を出力して、リードデータ保持部１３に保持されているリードデータを以後の各ステージで先着順にＣＰＵへ転送する。

前記フロー第５段の判断で、ＣＰＵからライトストローブがアサートされている場合、フローは右方へ分岐して、その間制御手段１２はリードデータ保持部１３にドライブ信号を出力せず、またＣＰＵへもリードストローブ信号を出力せずに、リードデータ保持部１３の内容をそのまま保持させて、データバス１０によりライトデータを転送させ、ライトストローブがネゲートされてデータバス１０が空くステージまで待ったのち、リードデータを先着順にＣＰＵへ転送する。

第３図は上記のインタリーブ制御の一例を示すタイミングチャートで、サイクル及びステージの設定と、書込み／読出しのタイミングは第６図の場合と同じである。

第３図（ａ）は連続リードのタイミングを示す図である。第１サイクルの第１ステージ（Ｓ１）でＣＰＵ１がリードサイクルを起動すると、制御手段１２が２ステージ後にクロック信号を発し、メモリバンクからのリードデータをリードデータ保持部１３に保持させ、３ステージ後の第４ステージ（Ｓ４）にドライブ信号とリードストローブ信号を出力して、ＣＰＵ１へリードデータを転送する。ＣＰＵ２以下も同様にメモリリードする。

第３図（ｂ）は連続ライトのタイミングを示す図である。各ＣＰＵは、ライト起動をかけたステージでライトストローブ信号を出力し、次のステージでライトデータを転送する。

第３図（ｃ）は、ライトとリードとが混在するタイミングを示す図である。ＣＰＵ１とＣＰＵ２の要求したリードデータは、それぞれ２ステージ

後にはリードデータ保持部１３に保持されるが、第３ステージ（Ｓ３）及び第４ステージ（Ｓ４）でライトストローブ信号が出力されているので、ドライブ信号とリードストローブ信号は、第２サイクルの第２ステージ（Ｓ２′）及び第３ステージ（Ｓ３′）まで待たされ、リードデータの転送は２ステージ分待たされる。しかし、従来リード又はライトが１サイクル（４ステージ）分待たされていたのに比較すると、処理は高速化されている。

ところで、あるＣＰＵがライトを起動しようとしたとき、そのライトデータの転送とそれ以前に起動された別なＣＰＵのリードデータの転送とが重なる場合に、本発明ではライトデータの転送が優先されるが、仮にリードを起動した方のＣＰＵの優先順位がライトを起動したＣＰＵの優先順位よりも高い場合がある。このような場合、ＣＰＵの優先度をインタリーブ制御よりも優先させるため、本実施例では、ＣＰＵ５～８の上位にＣＰＵ優先度処理部１５が配設されている。

第４図は、本発明のアクセス権決定手順の一例を示すフローチャートである。第４図に示すように、まずＣＰＵ優先度処理部１５が、ＣＰＵの優先順位に従って、アクセスを起動するＣＰＵｉを決定し、そのアクセスがリードであれば無条件にリードサイクルを起動させる。当該ＣＰＵｉのアクセスがライトの場合、次ステージが他のＣＰＵｊのリードデータ転送のタイミングであるか、又はリードデータ転送を待っているＣＰＵｊが存在すれば、ＣＰＵｉの優先順位をＣＰＵｊの優先順位と比較する。それ以外は、ライトサイクルを起動する。前記比較の結果、当該ＣＰＵｉが優先度を有している場合も、ライトサイクルを起動する。比較の結果、次ＣＰＵｊの優先順位が高い場合はライトサイクルの起動を１サイクル待機させる。即ち、インタリーブ制御によるＣＰＵｉのライト優先権よりもＣＰＵｊのＣＰＵ優先度を重視した決定となる。

（発明の効果）

以上述べたように、本発明によれば、ＣＰＵ側の待ち時間を可能な限り短縮し、高速かつ効率的な処理を実現するメモリ制御方式を提供することができる。

14；ライトデータバッファ、

15；プロセッサ（ＣＰＵ）優先度処理部。

代理人　弁理士　井桁貞一

4．図面の簡単な説明

第1図は本発明の一実施例の構成図、

第2図は本発明のフローチャート、

第3図は実施例のタイミングチャート、

第4図は本発明の別な実施例のフローチャート、

第5図は従来例の構成図、

第6図は従来例のタイミングチャートである。

1～4；メモリバンク、

5～8；プロセッサ（ＣＰＵ）、

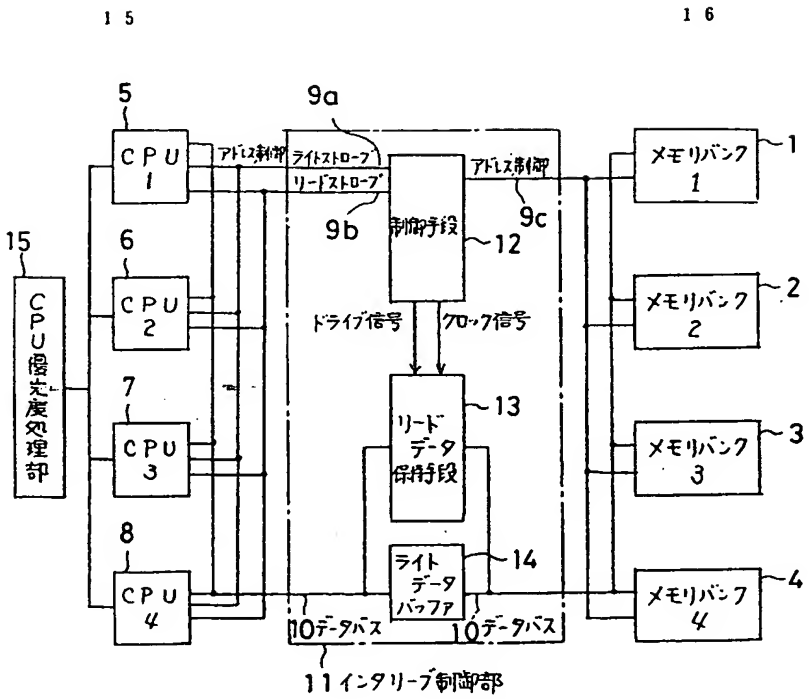
9；アドレスバス、

10；データバス、

11；インタリーブ制御部、

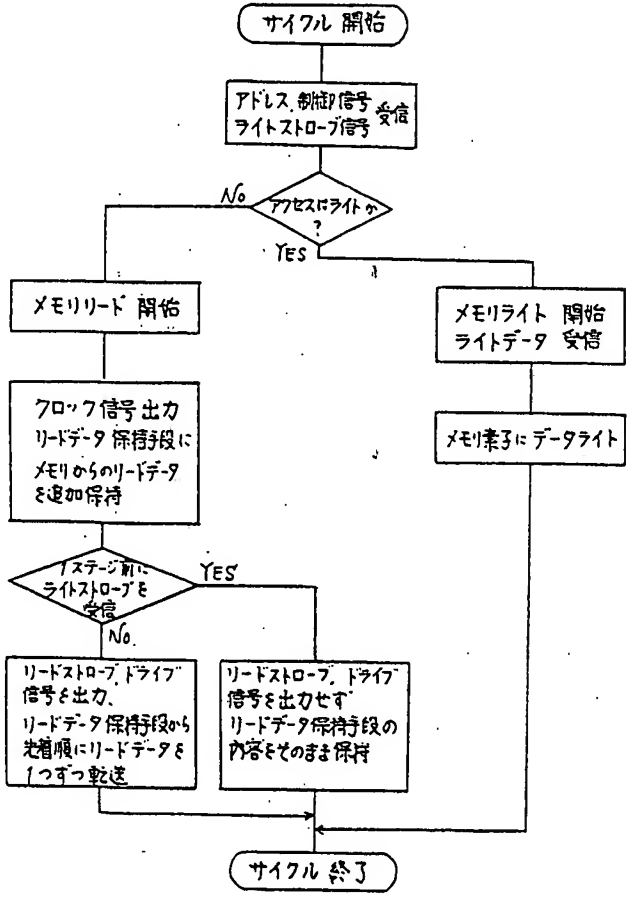
12；制御手段、

13；リードデータ保持手段、



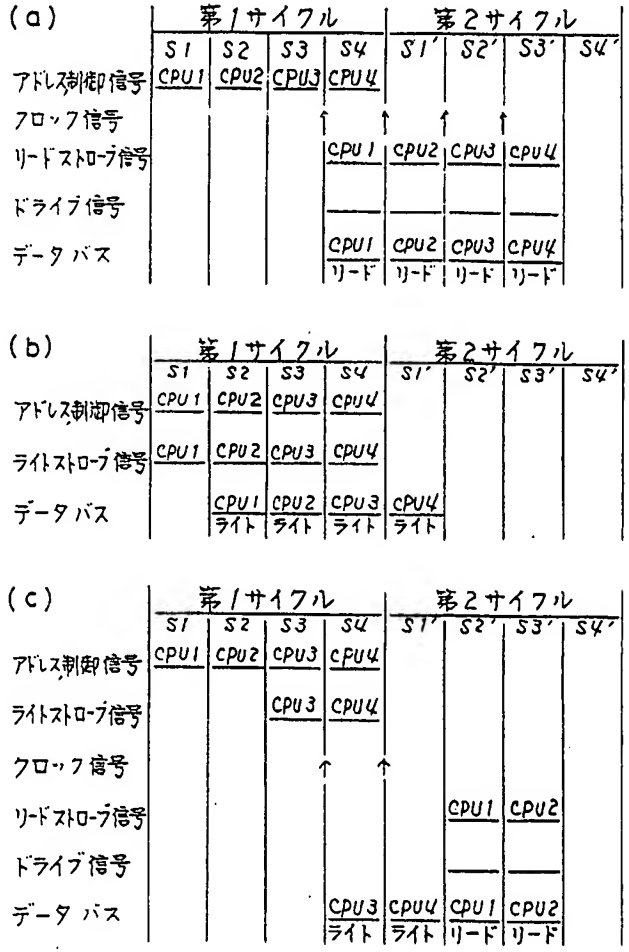
本発明の一実施例の構成図

第1図



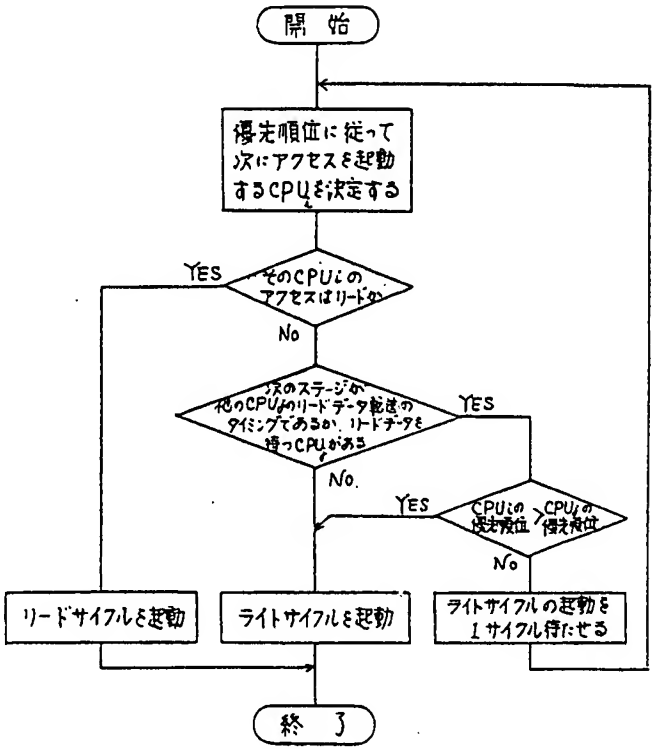
インタリーブ制御部の手順のフローチャート

第2図



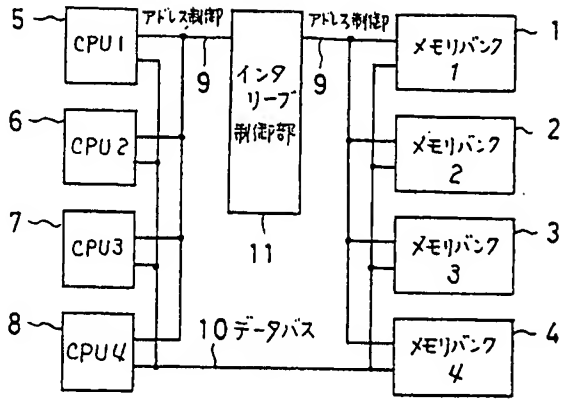
本発明の一実施例のタイミングチャート

第3図



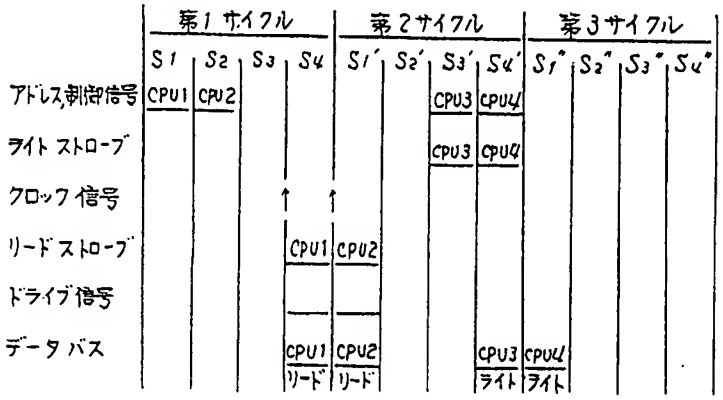
アクセス権決定手順のフローチャート

第4図



従来例の構成図

第5図



従来例のタイミングチャート

第6図